FAST FOURIER TRANSFORM (FFT) BUTTERFLY CALCULATIONS IN TWO CYCLES

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of U.S. Patent Application No. 09/587,617, filed June 5, 2000, which is hereby incorporated by reference.

BACKGROUND OF THE INVENTION

[0002] A digital signal processor (DSP) is a computer that is designed to optimize digital signal processing tasks. A non-exhaustive list of examples of such processing tasks includes Fast Fourier Transform (FFT) calculations, digital filters, image processing, and speech recognition.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] Embodiments of the invention are illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like reference numerals indicate corresponding, analogous or similar elements, and in which:

[0004] FIG. 1 is a simplified block diagram illustration of an exemplary digital signal processor (DSP) to perform Fast Fourier Transform (FFT) calculations, according to an embodiment of the invention;

[0005] FIG. 2 is a tabular illustration of the contents of registers of the exemplary DSP of FIG. 1 over several cycles; and

[0006] FIG. 3 is another tabular illustration of the contents of registers of the exemplary DSP of FIG. 1 over several cycles.

[0007] It will be appreciated that for simplicity and clarity of illustration, elements shown in the figures have not necessarily been drawn to scale. For example, the dimensions of some of the elements may be exaggerated relative to other elements for clarity.

DETAILED DESCRIPTION OF THE INVENTION

[0008] In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of embodiments of the invention. However it will be understood by those of ordinary skill in the art that the embodiments of the invention may be practiced without these specific details. In other instances, well-known methods, procedures, components and circuits have not been described in detail so as not to obscure the embodiments of the invention.

[0009] FIG. 1 is a simplified block diagram illustration of an exemplary digital signal processor (DSP) 2 to perform Fast Fourier Transform (FFT) calculations, according to an embodiment of the invention. DSP 2 may perform other calculations, but these are not described so as not to obscure the description of the embodiments of the invention. DSP 2 may include two three-input arithmetic logic units (ALU) 10 and 12, each capable of receiving three inputs and performing any combination of addition and subtraction on the three inputs in response to program instructions to yield a combined result. DSP 2 may also include multipliers 14 and 16, labeled *MUL1* and *MUL2*, to perform multiplication on real and imaginary sinusoidal data inputs B_R and B_I and coefficients W_R and W_I using conventional techniques. Results from multipliers 14 and 16 may be stored in registers 18 and 20 respectively, labeled *P0* and *P1*, from which the results may then be input to ALUs 10 and 12.

[0010] DSP 2 may also include two registers 22 and 24, labeled *Zr0* and *Zr1*, to receive real cosinusoidal data input A_R, and two registers 26 and 28, labeled *Zi0* and *Zi1*, to receive imaginary cosinusoidal data input A_I. DSP 2 may also include a multiplexer 30 to selectably provide data from registers *Zr0*, *Zr1* and *Zi1* to ALUs 10 and 12. DSP 2 may optionally concatenate a rounding constant C to the multiplexed data, shown at reference numeral 35, to form a low-ordered portion of the concatenated input to ALUs 10 and 12.

[0011] DSP 2 may also include two registers 34 and 36, labeled *A0* and *A1*, to receive output from ALU 10, and two registers 38 and 40, labeled *A2* and *A3*, to receive output from ALU 12. DSP 2 may also include a register 42, labeled *A0hp*, to receive a high-ordered portion of the data stored in *A0*, and a register 44, labeled *A2hp*, to receive a high-ordered portion of the data stored in *A2*.

[0012] DSP 2 may also include a multiplexer 46 to selectably provide data from A0hp or A2hp. DSP 2 may also include a multiplexer 48 to selectably provide data from A1 or A3.

[0013] DSP 2 may include additional components that are not shown in FIG. 1 so as not to obscure the description of embodiments of the invention.

[0014] An exemplary FFT butterfly calculation will now be described with respect to FIG. 1 and FIG. 2, which is a tabular illustration of the contents of registers of DSP 2 over several cycles.

[0015] Each FFT butterfly calculation, indexed by k, is to result in four outputs:

$$\mathrm{OUT0}[k] = A_R[k] + B_R[k]*W_R[k] - B_I[k]*W_I[k]$$

$$\mathrm{OUT1}[k] = A_I[k] + B_R[k]*W_I[k] + B_I[k]*W_R[k]$$

$$\mathrm{OUT2}[k] = A_R[k] - B_R[k]*W_R[k] + B_I[k]*W_I[k]$$

$$\mathrm{OUT3}[k] = A_I[k] - B_R[k]*W_I[k] - B_I[k]*W_R[k]$$

where, if the optional rounding constant is used, then $A_R[k]$ ($A_I[k]$) is replaced by $A_R[k]*C$ ($A_I[k]*C$) in the equations above, and the following description will demonstrate one example of how these four outputs for a particular butterfly calculation may be calculated in two cycles.

[0016] In an exemplary initial state, registers *Zr0* and *Zi0*, and registers *Zr1* and *Zi1* may store the first real cosinusoidal data input ($A_R[1]$) and the first imaginary cosinusoidal data input ($A_I[1]$), respectively, register *P0* may store the product of the first real sinusoidal data input ($B_R[1]$) and the first real coefficient ($W_R[1]$), and register *P1* may store the product of the first imaginary sinusoidal data input ($B_I[1]$) and the first imaginary coefficient ($W_I[1]$).

CYCLE #1

[0017] During a first cycle, labeled CYCLE #1, the following actions may occur:

a) multiplexer 30 may retrieve the contents of *Zr1* ($A_R[1]$), the rounding constant C may optionally be concatenated to that value, and the possibly concatenated output of multiplexer 30 may be provided to ALUs 10 and 12; ALU 10 may add the possibly concatenated output to the contents of register *P0* ($B_R[1]*W_R[1]$) and subtract therefrom the contents of register *P1* ($B_I[1]*W_I[1]$) and store the result (OUT0[1]) in register *A0*; ALU 12 may add the possibly concatenated output to the contents of register *P1* and subtract therefrom the contents of register *P0* and store the result (OUT2 [1]) in register *A2*;

b) registers *Zr0* and *Zi0* may receive the real and imaginary cosinusoidal data inputs for the second FFT butterfly ($A_R[2]$ and $A_I[2]$, respectively); and

c) multiplier *MUL1* may multiply the first real sinusoidal data input ($B_R[1]$) with the first imaginary coefficient ($W_I[1]$) and store the product in register *P0*, and multiplier *MUL2* may multiply the first imaginary sinusoidal data input ($B_I[1]$) with the first real coefficient ($W_R[1]$) and store the product in register *P1*.

CYCLE #2

[0018] During a second cycle, labeled CYCLE #2, the following actions may occur:

a) a high-ordered portion of registers *A0* and *A2* (containing outputs of the first FFT butterfly calculation) may be copied to registers *A0hp* and *A2hp*, respectively;

b) multiplexer 30 may retrieve the contents of *Zi1* ($A_I[1]$), the rounding constant C may optionally be concatenated to that value, and the possibly concatenated output of multiplexer 30 may be provided to ALUs 10 and 12; ALU 10 may add the possibly concatenated output to the contents of register *P0* ($B_R[1]*W_I[1]$) and the contents of register *P1* ($B_I[1]*W_R[1]$) and store the result (OUT1[1]) in register *A1*; ALU 12 may subtract both the contents of register *P0* and the contents of register *P1* from the possibly concatenated output and store the result (OUT3 [1]) in register *A3*; and

c) multiplier *MUL1* may multiply the second real sinusoidal data input (B_R [2]) with the second real coefficient (W_R [2]) and store the product in register *P0*, and multiplier *MUL2* may multiply the second imaginary sinusoidal data input (B_I [2]) with the second imaginary coefficient (W_I [2]) and store the product in register *P1*; and

d) the contents of registers *Zr0* (A_R [2]) and *Zi0* (A_I [2]) may be input to registers *Zr1* and *Zi1*.

[0019] It should be noted that at the end of CYCLE #2, the four outputs of the first FFT butterfly calculation, (OUT0 [1], OUT1 [1], OUT2 [1], OUT3 [1]) have been calculated and are stored in registers *A0* (and *A0hp*), *A1*, *A2* (and *A2hp*) and *A3*, respectively.

CYCLE #3

[0020] During a third cycle, labeled CYCLE #3, the following actions may occur:

a) multiplexer 30 may retrieve the contents of *Zr1* (A_R [2]), the rounding constant C may optionally be concatenated to that value, and the possibly concatenated output of multiplexer 30 may be provided to ALUs 10 and 12; ALU 10 may add the possibly concatenated output to the contents of register *P0* (B_R [2] *W_R [2]) and subtract therefrom the contents of register *P1* (B_I [2] *W_I [2]) and store the result (OUT0 [2]) in register *A0*; ALU 12 may add the possibly concatenated output to the contents of register *P1* and subtract therefrom the contents of register *P0* and store the result (OUT2 [2]) in register *A2*;

b) registers *Zr0* and *Zi0* may receive the real and imaginary cosinusoidal data inputs for the third FFT butterfly (A_R [3] and A_I [3] , respectively); and

c) multiplier *MUL1* may multiply the second real sinusoidal data input ($B_R[2]$) with the second imaginary coefficient ($W_I[2]$) and store the product in register *P0*, and multiplier *MUL2* may multiply the second imaginary sinusoidal data input ($B_I[2]$) with the second real coefficient ($W_R[2]$) and store the product in register *P1*.

CYCLE #4

[0021] During a fourth cycle, labeled CYCLE #4, the following actions may occur:

a) a high-ordered portion of registers *A0* and *A2* (containing outputs of the second FFT butterfly calculation) may be copied to registers *A0hp* and *A2hp*, respectively;

b) multiplexer 30 may retrieve the contents of *Zi1* ($A_I[2]$), the rounding constant C may optionally be concatenated to that value, and the possibly concatenated output of multiplexer 30 may be provided to ALUs 10 and 12; ALU 10 may add the possibly concatenated output to the contents of register *P0* ($B_R[2]*W_I[2]$) and the contents of register *P1* ($B_I[2]*W_R[2]$) and store the result (OUT1[2]) in register *A1*; ALU 12 may subtract both the contents of register *P0* and the contents of register *P1* from the possibly concatenated output and store the result (OUT3[2]) in register *A3*; and

c) multiplier *MUL1* may multiply the third real sinusoidal data input ($B_R[3]$) with the third real coefficient ($W_R[3]$) and store the product in register *P0*, and multiplier *MUL2* may multiply the third imaginary sinusoidal data input ($B_I[3]$) with the third imaginary coefficient ($W_I[3]$) and store the product in register *P1*; and

d) the contents of registers *Zr0* ($A_R[3]$) and *Zi0* ($A_I[3]$) may be input to registers *Zr1* and *Zi1*, respectively.

[0022] It should be noted that at the end of CYCLE #4, the four outputs of the second FFT butterfly calculation, (OUT0[2], OUT1[2], OUT2[2], OUT3[2]) have been calculated and are stored in registers *A0* (and *A0hp*), *A1*, *A2* (and *A2hp*) and *A3*, respectively.

CYCLE #5

[0023] During a fifth cycle, labeled CYCLE #5, the following actions may occur:

a) multiplexer 30 may retrieve the contents of *Zr1* ($A_R[3]$), the rounding constant C may optionally be concatenated to that value, and the possibly concatenated output of multiplexer 30 may be provided to ALUs 10 and 12; ALU 10 may add the possibly concatenated output to the contents of register *P0* ($B_R[3]*W_R[3]$) and subtract therefrom the contents of register *P1* ($B_I[3]*W_I[3]$) and store the result (OUT0[3]) in register *A0*; ALU 12 may add the possibly concatenated output to the contents of register *P1* and subtract therefrom the contents of register *P0* and store the result (OUT2[3]) in register *A2*;

b) registers *Zr0* and *Zi0* may receive the real and imaginary cosinusoidal data inputs for the fourth FFT butterfly ($A_R[4]$ and $A_I[4]$, respectively); and

c) multiplier *MUL1* may multiply the third real sinusoidal data input ($B_R[3]$) with the third imaginary coefficient ($W_I[3]$) and store the product in register *P0*, and multiplier *MUL2* may multiply the third imaginary sinusoidal data input ($B_I[3]$) with the third real coefficient ($W_R[3]$) and store the product in register *P1*.

CYCLE #6

[0024] During a sixth cycle, labeled CYCLE #6, the following actions may occur:

a) a high-ordered portion of registers *A0* and *A2* (containing outputs of the third FFT butterfly calculation) may be copied to registers *A0hp* and *A2hp*, respectively;

b) multiplexer 30 may retrieve the contents of *Zi1* ($A_I[3]$), the rounding constant C may optionally be concatenated to that value, and the possibly concatenated output of multiplexer 30 may be provided to ALUs 10 and 12; ALU 10 may add the possibly concatenated output to the contents of register *P0* (B_R[3]*W_I[3]) and the contents of register *P1* (B_I[3]*W_R[3]) and store the result (OUT1[3]) in register *A1*; ALU 12 may subtract both the contents of register *P0* and the contents of register *P1* from the possibly concatenated output and store the result (OUT3 [3]) in register *A3*; and

c) multiplier *MUL1* may multiply the fourth real sinusoidal data input (B_R [4]) with the fourth real coefficient (W_R [4]) and store the product in register *P0*, and multiplier *MUL2* may multiply the fourth imaginary sinusoidal data input (B_I [4]) with the fourth imaginary coefficient (W_I [4]) and store the product in register *P1*; and

d) the contents of registers *Zr0* (A_R [4]) and *Zi0* (A_I [4]) may be input to registers *Zr1* and *Zi1*, respectively.

[0025] It should be noted that at the end of CYCLE #6, the four outputs of the third FFT butterfly calculation, (OUT0[3], OUT1[3], OUT2[3], OUT3[3]) have been calculated and are stored in registers *A0* (and *A0hp*), *A1*, *A2* (and *A2hp*) and *A3*, respectively.

SUBSEQUENT CYCLES

[0026] The actions of CYCLES #7 and #9 are similar to those of CYCLES #1, #3, and #5, while the actions of CYCLE #8 are similar to those of CYCLES #2, #4 and #6. Subsequent cycles are performed until all the input data has been fully processed.

DATA PROPAGATION

[0027] Consequently, the data propagation in the structure shown in FIG. 1 may be considered as follows:

a) Registers *Zr0* and *Zi0* receive the real and imaginary cosinusoidal data inputs for the FFT butterfly (A_R and A_I, respectively) in each “first cycle” (CYCLE #1, CYCLE #3, etc.), and maintain their values in each "second cycle" (CYCLE #2, CYCLE #4, etc.).

b) Registers *Zr1* and *Zi1* receive the contents of registers *Zr0* and *Zi0* respectively in each "second cycle" and maintain their values in each "first cycle".

c) In each "first cycle", multiplier *MUL1* multiplies the real sinusoidal data input (B_R) with the imaginary coefficient (W_I) and stores the product in register *P0*, and multiplier *MUL2* multiplies the imaginary sinusoidal data input (B_I) with the real coefficient (W_R) and stores the product in register *P1*. In each "second cycle", multiplier *MUL1* multiplies the real sinusoidal data input (B_R) with the real coefficient (W_R) and stores the product in register *P0*, and multiplier *MUL2* multiplies the imaginary sinusoidal data input (B_I) with the imaginary coefficient (W_I) and stores the product in register *P1*.

d) In each "first cycle" multiplexer 30 may retrieve the contents of *Zr1* (A_R), the rounding constant C may optionally be concatenated to that value, and the possibly concatenated output of multiplexer 30 may be provided to ALUs 10 and 12; ALU 10 may add the possibly concatenated output to the contents of register *P0* (B_R*W_R) and subtract therefrom the contents of register *P1* (B_I*W_I) and store the result (OUT0) in register *A0*. ALU 12 may add the possibly concatenated output to the contents of register *P1* and subtract therefrom the contents of register *P0* and store the result (OUT2) in register *A2*. Registers *A0* and *A2* maintain their values in each "second cycle".

e) In each "second cycle", a high-ordered portion of registers *A0* and *A2* may be copied to registers *A0hp* and *A2hp*, respectively. Registers *A0hp* and *A2hp* maintain their values in each "first cycle".

f) In each "second cycle" multiplexer 30 may retrieve the contents of *Zi1* (A_I), the rounding constant C may optionally be concatenated to that value, and the possibly concatenated output of multiplexer 30 may be provided to ALUs 10 and 12; ALU 10 may add the possibly concatenated output to the contents of register *P0* (B_R*W_I) and the contents of register *P1* (B_I*W_R) and store the result (OUT1) in register *A1*. ALU 12 may subtract both the contents of register *P0* and the contents of register *P1* from the possibly concatenated output and store the result (OUT3) in register *A3*. Registers *A1* and *A3* maintain their values in each "first cycle".

READING FFT CALCULATION RESULTS TO MEMORY

[0028] As mentioned hereinabove, DSP 2 may include multiplexer 46 to selectably provide data from registers *A0hp* or *A2hp*, and multiplexer 48 to selectably provide data from registers *A1* or *A3*. Therefore, in any given cycle, data may be read from *A0hp* and *A1*, or from *A0hp* and *A3*, or from *A2hp* and *A1*, or from *A2hp* and *A3*. In the following examples, data is read from registers *A0hp* and *A1* in one cycle and from registers *A2hp* and *A3* in the next cycle.

[0029] The reading of the FFT calculation results OUT0[k] and OUT1[k] during a "first cycle" (CYCLE #3, #5, #7, #9, etc.) is indicated in FIG. 2 by diagonal lines, where the values read are the values in registers *A0hp* and *A1* at the end of the previous "first cycle" (CYCLE #2, #4, #6, #8, etc., respectively).

[0030] The reading of the FFT calculation results OUT2[k] and OUT3[k] during a "second cycle" (CYCLE #4, #6, #8, etc.) is indicated in FIG. 2 by diagonal lines, where the values read are the values in registers *A2hp* and *A3* at the end of the previous "first cycle" (CYCLE #3, #5, #7, etc., respectively).

[0031] Consequently, it should be noted that all four FFT calculation results from a single butterfly may be read in two cycles.

[0032] FIG. 3 is another tabular illustration of the contents of registers of the exemplary DSP of FIG. 1 over several cycles. FIG. 3 is identical to FIG. 2, except that FIG. 3 shows an alternate manner for reading the FFT calculation results.

[0033] The reading of the FFT calculation results OUT2[k] and OUT3[k] during a "first cycle" (CYCLE #3, #5, #7, etc.) is indicated in FIG. 3 by diagonal lines, where the values read are the values in registers *A2hp* and *A3* at the end of the previous "second cycle" (CYCLE #2, #4, #6, etc., respectively).

[0034] The reading of the FFT calculation results OUT0[k] and OUT1[k] during a "second cycle" (CYCLE #4, #6, #8, etc.) is indicated in FIG. 3 by diagonal lines, where the values read are the values in registers *A0hp* and *A1* at the end of the previous "first cycle" (CYCLE #3, #5, #7, etc., respectively).

[0035] Consequently, it should be noted that all four FFT calculation results from a single butterfly may be read in two cycles.

[0036] It should also be noted that other manners for reading the FFT calculation results in two or more cycles are also applicable to the embodiments of the present invention. For example, the manner shown in FIG. 2 may be used for some pairs of consecutive cycles and the manner shown in FIG. 3 may be used for other pairs of consecutive cycles.

[0037] While certain features of the invention have been illustrated and described herein, many modifications, substitutions, changes, and equivalents will now occur to those of ordinary skill in the art. It is, therefore, to be understood that the appended claims are intended to cover all such modifications and changes as fall within the true spirit of the invention.